LAPORAN TUGAS KECIL 1
IF2211 STRATEGI ALGORITMA
"PENYELESAIAN PERMAINAN KARTU 24 DENGAN ALGORITMA BRUTE FORCE"

Disusun oleh:
Moh. Aghna Maysan Abyan (13521076)

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

SEMESTER II TAHUN AJARAN 2022/2023

ALGORITMA BRUTE FORCE

Metode penyelesaian dengan algoritma *Brute Force* adalah dengan mengecek tiap-tiap kemungkinan yang ada, hingga tercapai suatu hasil yang diinginkan. Pada tugas kecil kali ini, program ditugaskan untuk mengecek tiap operasi yang memungkinkan dari empat input angka dengan operasi +, -, *, /, (, dan). Tiap kemungkinan akan dihitung satu per satu, dan jika hasilnya berupa 24, bentuk operasinya akan di *print* ke layar.

Pada program sendiri ada dua pilihan masukan, antara *random* (pilihan kartu diacak oleh program) atau pengguna dapat memasukan angka/huruf yang diinginkan, dengan pilihannya adalah: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, dan K.

Untuk contoh saja, misalkan pengguna melakukan masukan empat kartu yaitu 3, 5, 7, dan 8, maka program akan menghitung tiap kemungkinan operasi dari angka 3, 5, 7, dan 8. Salah satu kemungkinan operasi yang dapat menghasilkan 24 adalah 3-((5-8)*7) dengan urutan operasi sebagai berikut:

- 3-((5-8)*7)
- 3-(-3*7)
- 3-(-21)
- 24

Terbukti bahwa operasi diatas dapat menghasilkan 24. Sehingga, program akan meng-*print* operasi 3-((5-8)*7) pada hasil akhir program.

Pada program yang telah saya buat, ditemukan adanya 42 kombinasi yang memungkinkan dari angka 3, 5, 7, dan 8 yang menghasilkan nilai 24. Sehingga, terdapat total 42 operasi yang harus di-*print* pada akhiran program.

SOURCE CODE

Saya menggunakan bahasa pemrograman Java untuk tugas kecil ini. *Source code* dari program yang telah saya buat adalah sebagai berikut:

File: brute.java

Class card

Mengubah nilai string kartu ke integer (A jadi 1, 2-10 sama aja, J jadi 11, Q jadi 12, dan K jadi 13).

```java
class card {
    public static int convertCardToInt(String rank) {
        switch(rank){
            case "A":
                return 1;
            case "2":
                return 2;
            case "3":
                return 3;
            case "4":
                return 4;
            case "5":
                return 5;
            case "6":
                return 6;
            case "7":
                return 7;
            case "8":
                return 8;
            case "9":
                return 9;
            case "10":
                return 10;
            case "J":
                return 11;
            case "Q":
                return 12;
            case "K":
                return 13;
            default:
                return 0;
        }
    }
}
```

Class DuaPuluhEmpat

Mengecek properti/nilai tiap kartu, dan mencari solusinya.

```java
class DuaPuluhEmpat {
    private static final List<IntBinaryOperator> operators = new ArrayList<IntBinaryOperator>();
    private static final List<Character> opnames = new ArrayList<Character>();

    public DuaPuluhEmpat() {
        operators.add((a, b) -> a + b);
        operators.add((a, b) -> a - b);
        operators.add((a, b) -> a * b);
        operators.add((a, b) -> (b != 0 && a % b == 0) ? a / b : Integer.MIN_VALUE);
        operators.add((a, b) -> (a == 0 || a == 1 || b < 9) ? (int) Math.pow(a, b) : Integer.MIN_VALUE);
        opnames.add(e: '+');
        opnames.add(e: '-');
        opnames.add(e: '*');
        opnames.add(e: '/');
    }

    public List<String> solusi(String[] num) {
        List<Integer> digits = new ArrayList<Integer>();
        digits.add(card.convertCardToInt(num[0]));
        digits.add(card.convertCardToInt(num[1]));
        digits.add(card.convertCardToInt(num[2]));
        digits.add(card.convertCardToInt(num[3]));
        ArrayList<String> solutions = new ArrayList<String>();
        for (int a = 0; a < 4; a++) {
            for (int b = 0; b < 4; b++) {
                for (int c = 0; c < 4; c++) {
                    for (int d = 0; d < 4; d++) {
                        if ((a == b) || (a == c) || (a == d) || (b == c) || (b == d) || (c == d)) {
                            continue;
                        }
```

```java
                        for (int i = 0; i < 4; i++) {
                            for (int j = 0; j < 4; j++) {
                                for (int k = 0; k < 4; k++) {
                                    if (operators.get(j).applyAsInt(operators.get(i).applyAsInt(digits.get(a), digits.get(b)), operators.get(k).applyAsInt(digits.get(c), digits.get(d))) == 24) {
                                        String sol_str = "(%d %s %d) %s (%d %s %d)";
                                        sol_str = String.format(sol_str, digits.get(a), opnames.get(i), digits.get(b), opnames.get(j), digits.get(c), opnames.get(k), digits.get(d));
                                        solutions.add(sol_str);
                                    }
                                    if (operators.get(k).applyAsInt(operators.get(j).applyAsInt(operators.get(i).applyAsInt(digits.get(a), digits.get(b)), digits.get(c)), digits.get(d)) == 24) {
                                        String sol_str = "((%d %s %d) %s %d) %s %d";
                                        sol_str = String.format(sol_str, digits.get(a), opnames.get(i), digits.get(b), opnames.get(j), digits.get(c), opnames.get(k), digits.get(d));
                                        solutions.add(sol_str);
                                    }
                                    if (operators.get(k).applyAsInt(operators.get(i).applyAsInt(digits.get(a), operators.get(j).applyAsInt(digits.get(b), digits.get(c))), digits.get(d)) == 24) {
                                        String sol_str = "(%d %s (%d %s %d)) %s %d";
                                        sol_str = String.format(sol_str, digits.get(a), opnames.get(i), digits.get(b), opnames.get(j), digits.get(c), opnames.get(k), digits.get(d));
                                        solutions.add(sol_str);
                                    }
                                    if (operators.get(i).applyAsInt(digits.get(a), operators.get(k).applyAsInt(operators.get(j).applyAsInt(digits.get(b), digits.get(c)), digits.get(d))) == 24) {
                                        String sol_str = "%d %s ((%d %s %d) %s %d)";
                                        sol_str = String.format(sol_str, digits.get(a), opnames.get(i), digits.get(b), opnames.get(j), digits.get(c), opnames.get(k), digits.get(d));
                                        solutions.add(sol_str);
                                    }
                                    if (operators.get(i).applyAsInt(digits.get(a), operators.get(j).applyAsInt(digits.get(b), operators.get(k).applyAsInt(digits.get(c), digits.get(d)))) == 24) {
                                        String sol_str = "%d %s (%d %s (%d %s %d))";
                                        sol_str = String.format(sol_str, digits.get(a), opnames.get(i), digits.get(b), opnames.get(j), digits.get(c), opnames.get(k), digits.get(d));
                                        solutions.add(sol_str);
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
        System.out.println(solutions.size() + " Solutions found");
        if(solutions.size()>0) {
            for (String solution : solutions) {
                System.out.println(solution);
            }
        } else {
            System.out.println(x: "Tidak ada solusi");
        }
    return solutions;
    }
}

public class brute {
}
```

File: main.java

Class main

Tempat untuk menjalankan program, terdapat UI program serta inputnya.

```java
import java.io.*;
import java.util.*;

public class main{
    public static List<String> randomList(){
        List<String> arrayCard = new ArrayList<>();
        arrayCard.add(e: "A");arrayCard.add(e: "2");arrayCard.add(e: "3");arrayCard.add(e: "4");
        arrayCard.add(e: "5");arrayCard.add(e: "6");arrayCard.add(e: "7");arrayCard.add(e: "8");
        arrayCard.add(e: "9");arrayCard.add(e: "10");arrayCard.add(e: "J");arrayCard.add(e: "Q");
        arrayCard.add(e: "K");
        Collections.shuffle(arrayCard);
        return arrayCard.subList(fromIndex: 0, toIndex: 4);
    }

    public static void createFile(String filename, List<String> list){
        try{
            BufferedWriter bw = new BufferedWriter(new FileWriter(filename));
            for (String s: list){
                bw.write(s);
                bw.newLine();
            }
            System.out.println(x: "File created and write succesfully");
            bw.close();
        }
        catch (IOException e){
            System.out.println(x: "An Error Has Occured");
            e.printStackTrace();
        }
    }

```

```java
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        Scanner masukan = new Scanner(System.in);
        String kartu;

        List<String> card_input = new ArrayList<String>();
        boolean valid;
        int inputuser = 0, n = 0;
        while(inputuser != 1 && inputuser != 2) {
            System.out.println(x: "Pilih masukkan yang diinginkan: ");
            System.out.println(x: "1. Random");
            System.out.println(x: "2. Keyboard Sendiri");


            inputuser = input.nextInt();
            if (inputuser == 1){
                card_input = randomList();
            }
            else if (inputuser == 2){
            System.out.println(x: "Masukkan 4 angka/huruf (A,2,3,4,5,6,7,8,9,10,J,Q,K)");
            while (n < 4){
                valid = false;
                while (!valid){
                    kartu = masukan.nextLine();
                    if (card.convertCardToInt(kartu) == 0){
                        System.out.println(x: "Masukan Tidak Sesuai");
                        }
                    else{
                        card_input.add(kartu);
                        valid = true;
                        n++;
                        }
                    }
                }
            }
        }
```

```java
            String[] cards = (String[]) card_input.toArray(new String[0]);
            DuaPuluhEmpat game = new DuaPuluhEmpat();

            long startTime = System.currentTimeMillis();
            List<String> hasil = game.solusi(cards);
            long endTime = System.currentTimeMillis();
            long totalTime = endTime - startTime;
            String nama_file;

            System.out.println(x: "Apakah ingin menyimpan solusi? (Y/N)");
            char final2 = masukan.next().charAt(index: 0);
            while(final2 != ('Y') && final2 !=('N')){
                if (final2 == ('Y')){
                    break;
                }
                else if (final2 == ('N')){
                    break;
                }
                System.out.println(x: "Apakah ingin menyimpan solusi? (Y/N)");
                final2 = masukan.next().charAt(index: 0);
            }
            Scanner nama = new Scanner(System.in);
            if (final2 == 'Y'){
                System.out.println(x: "Masukkan Nama File:");
                nama_file = nama.nextLine();
                nama_file = nama_file + ".txt";
                createFile(nama_file, hasil);
            }

            System.out.println("Execution time: " + totalTime + " ms");
            input.close();
            masukan.close();
            nama.close();
        }
    }
```

INPUT & OUTPUT

CATATAN: Pada *screenshot* dibawah, jumlah solusi yang ditampilkan terpotong sehingga jumlahnya tidak sesuai dengan “x solutions found.” Namun, semua solusi tertuang pada tiap file .txt yang tertera pada folder test di repository “Tucil1_13521076.” Pengguna bisa mencoba sendiri programnya untuk memastikan bahwa semua solusi tertulis pada file .txt.

1) 5 6 7 K

```
PS C:\Users\MSI\OneDrive\Dokumen\Kuliah\Kuliah Semester 4\Strategi Algoritma\Tucil1_13521076>  c:; cd 'c:\Users\MSI\OneDrive\Dokumen\Kuliah\Kuliah Semester 4\Strategi Algoritma\Tucil1_13521076'; & 'C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\MSI\AppData\Roaming\Code\User\workspaceStorage\9daae880b65f6bd9de10a9a066e81ec5\redhat.java\jdt_ws\Tucil1_13521076_5bf31e26\bin' 'main'
Pilih masukkan yang diinginkan:
1. Random
2. Keyboard Sendiri
2
Masukkan 4 angka/huruf (A,2,3,4,5,6,7,8,9,10,J,Q,K)
5
6
7
K
34 Solutions found
(5 * 6) + (7 - 13)
((5 * 6) + 7) - 13
((5 * 6) - 13) + 7
(5 * 6) - (13 - 7)
(5 * (13 - 7)) - 6
(6 * 5) + (7 - 13)
((6 * 5) + 7) - 13
((6 * 5) - 13) + 7
(6 * 5) - (13 - 7)
(6 * 7) - (5 + 13)
((13 + 7) / 5) * 6
((13 - 7) * 5) - 6
((13 + 7) * 6) / 5
Apakah ingin menyimpan solusi? (Y/N)
Y
Masukkan Nama File:
no1
File created and write succesfully
Execution time: 13 ms
PS C:\Users\MSI\OneDrive\Dokumen\Kuliah\Kuliah Semester 4\Strategi Algoritma\Tucil1_13521076>
```

2) A 7 9 J

```
PS C:\Users\MSI\OneDrive\Dokumen\Kuliah\Kuliah Semester 4\Strategi Algoritma\Tucil1_13521076>  c:; cd 'c:\Users\MSI\OneDrive\Dokumen\Kuliah\Kuliah Semester 4\Strategi Algoritma\Tucil1_13521076'; & 'C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\MSI\AppData\Roaming\Code\User\workspaceStorage\9daae880b65f6bd9de10a9a066e81ec5\redhat.java\jdt_ws\Tucil1_13521076_5bf31e26\bin' 'main'
Pilih masukkan yang diinginkan:
1. Random
2. Keyboard Sendiri
2
Masukkan 4 angka/huruf (A,2,3,4,5,6,7,8,9,10,J,Q,K)
A
7
9
J
4 Solutions found
(1 + 11) * (9 - 7)
(9 - 7) * (1 + 11)
(9 - 7) * (11 + 1)
(11 + 1) * (9 - 7)
Apakah ingin menyimpan solusi? (Y/N)
Y
Masukkan Nama File:
no2
File created and write succesfully
Execution time: 7 ms
PS C:\Users\MSI\OneDrive\Dokumen\Kuliah\Kuliah Semester 4\Strategi Algoritma\Tucil1_13521076>
```

3) 7 6 A 4

```
Pilih masukkan yang diinginkan:
1. Random
2. Keyboard Sendiri
2
Masukkan 4 angka/huruf (A,2,3,4,5,6,7,8,9,10,J,Q,K)
7
6
A
4
28 Solutions found
((7 + 1) - 4) * 6
(7 + (1 - 4)) * 6
((7 - 4) + 1) * 6
(7 - (4 - 1)) * 6
6 * ((7 + 1) - 4)
6 * (7 + (1 - 4))
6 * ((7 - 4) + 1)
6 * (7 - (4 - 1))
(6 + (1 / 7)) * 4
(6 - (1 / 7)) * 4
6 * ((1 + 7) - 4)
6 * (1 + (7 - 4))
6 * ((1 / 7) + 4)
6 * ((1 - 4) + 7)
6 * (1 - (4 - 7))
6 * (4 + (1 / 7))
6 * (4 - (1 / 7))
((1 / 7) + 6) * 4
((1 + 7) - 4) * 6
(1 + (7 - 4)) * 6
((1 / 7) + 4) * 6
((1 - 4) + 7) * 6
(1 - (4 - 7)) * 6
4 * (6 + (1 / 7))
4 * (6 - (1 / 7))
(4 + (1 / 7)) * 6
(4 - (1 / 7)) * 6
4 * ((1 / 7) + 6)
Apakah ingin menyimpan solusi? (Y/N)
Y
Masukkan Nama File:
no3
File created and write succesfully
Execution time: 23 ms
PS C:\Users\MSI\OneDrive\Dokumen\Kuliah\Kuliah Semester 4\Strategi Algoritma\Tucil1_13521076>
```

4) 2 3 4 K (random)

```
PS C:\Users\MSI\OneDrive\Dokumen\Kuliah\Kuliah Semester 4\Strategi Algoritma\Tucil1_13521076>  c:; cd 'c:\Users\MSI\OneDrive\Dokumen\Kuliah\Kuliah Semester 4\Strategi Algoritma\Tucil1_13521076'; & 'C:\Program Files\Java\jdk-18.0.2.1\bin\
java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\MSI\AppData\Roaming\Code\User\workspaceStorage\9daae880b65f6bd9de10a9a066e81ec5\redhat.java\jdt_ws\Tucil1_13521076_5bf31e26\bin' 'main'
Pilih masukkan yang diinginkan:
1. Random
2. Keyboard Sendiri
1
46 Solutions found
2 * ((13 + 3) - 4)
2 * (13 + (3 - 4))
(2 * (13 - 3)) + 4
2 * ((13 - 4) + 3)
2 * (13 - (4 - 3))
2 * ((3 + 13) - 4)
2 * (3 + (13 - 4))
2 * ((3 - 4) + 13)
2 * (3 - (4 - 13))
(2 * 4) + (13 + 3)
((2 * 4) + 13) + 3
(2 * 4) + (3 + 13)
((2 * 4) + 3) + 13
(13 + (2 * 4)) + 3
13 + ((2 * 4) + 3)
(13 + 3) + (2 * 4)
13 + (3 + (2 * 4))
((13 - 3) * 2) + 4
(13 + 3) + (4 * 2)
13 + (3 + (4 * 2))
((13 + 3) - 4) * 2
(13 + (3 - 4)) * 2
(13 + (4 * 2)) + 3
13 + ((4 * 2) + 3)
((13 - 4) + 3) * 2
(13 - (4 - 3)) * 2
(3 + (2 * 4)) + 13
3 + ((2 * 4) + 13)
(3 + 13) + (2 * 4)
3 + (13 + (2 * 4))
(3 + 13) + (4 * 2)
3 + (13 + (4 * 2))
((3 + 13) - 4) * 2
(3 + (13 - 4)) * 2
(3 + (4 * 2)) + 13
3 + ((4 * 2) + 13)
```

```
(3 + (4 * 2)) + 13
3 + ((4 * 2) + 13)
((3 - 4) + 13) * 2
(3 - (4 - 13)) * 2
4 + (2 * (13 - 3))
(4 * 2) + (13 + 3)
((4 * 2) + 13) + 3
4 - (2 * (3 - 13))
(4 * 2) + (3 + 13)
((4 * 2) + 3) + 13
4 + ((13 - 3) * 2)
4 - ((3 - 13) * 2)
Apakah ingin menyimpan solusi? (Y/N)
Y
Masukkan Nama File:
no4rand
File created and write succesfully
Execution time: 22 ms
PS C:\Users\MSI\OneDrive\Dokumen\Kuliah\Kuliah Semester 4\Strategi Algoritma\Tucil1_13521076>
```

5) A 3 10 J (random)

```
Pilih masukkan yang diinginkan:
1. Random
2. Keyboard Sendiri
1
196 Solutions found
(1 * 11) + (10 + 3)
((1 * 11) + 10) + 3
(1 * (11 + 10)) + 3
1 * ((11 + 10) + 3)
1 * (11 + (10 + 3))
(1 + (11 * 3)) - 10
1 + ((11 * 3) - 10)
(1 * 11) + (3 + 10)
((1 * 11) + 3) + 10
(1 * (11 + 3)) + 10
1 * ((11 + 3) + 10)
1 * (11 + (3 + 10))
(1 - 10) + (11 * 3)
1 - (10 - (11 * 3))
(1 * 10) + (11 + 3)
((1 * 10) + 11) + 3
(1 * (10 + 11)) + 3
1 * ((10 + 11) + 3)
1 * (10 + (11 + 3))
(1 - 10) + (3 * 11)
1 - (10 - (3 * 11))
(1 * 10) + (3 + 11)
((1 * 10) + 3) + 11
(1 * (10 + 3)) + 11
1 * ((10 + 3) + 11)
1 * (10 + (3 + 11))
(1 + (3 * 11)) - 10
1 + ((3 * 11) - 10)
(1 * 3) + (11 + 10)
((1 * 3) + 11) + 10
(1 * (3 + 11)) + 10
1 * ((3 + 11) + 10)
1 * (3 + (11 + 10))
(1 * 3) + (10 + 11)
((1 * 3) + 10) + 11
(1 * (3 + 10)) + 11
1 * ((3 + 10) + 11)
1 * (3 + (10 + 11))
1 * (3 + (10 + 11))
(11 + (1 * 10)) + 3
11 + ((1 * 10) + 3)
11 + (1 * (10 + 3))
(11 * 1) + (10 + 3)
((11 * 1) + 10) + 3
(11 / 1) + (10 + 3)
((11 / 1) + 10) + 3
(11 + (1 * 3)) + 10
11 + ((1 * 3) + 10)
11 + (1 * (3 + 10))
(11 * 1) + (3 + 10)
((11 * 1) + 3) + 10
(11 / 1) + (3 + 10)
((11 / 1) + 3) + 10
(11 + 10) + (1 * 3)
11 + (10 + (1 * 3))
((11 + 10) * 1) + 3
(11 + (10 * 1)) + 3
11 + ((10 * 1) + 3)
((11 + 10) / 1) + 3
(11 + (10 / 1)) + 3
11 + ((10 / 1) + 3)
(11 + 10) + (3 * 1)
((11 + 10) + 3) * 1
(11 + (10 + 3)) * 1
11 + ((10 + 3) * 1)
11 + (10 + (3 * 1))
(11 + 10) + (3 / 1)
((11 + 10) + 3) / 1
(11 + (10 + 3)) / 1
11 + ((10 + 3) / 1)
11 + (10 + (3 / 1))
(11 + 3) + (1 * 10)
11 + (3 + (1 * 10))
((11 + 3) * 1) + 10
(11 + (3 * 1)) + 10
11 + ((3 * 1) + 10)
((11 + 3) / 1) + 10
(11 + (3 / 1)) + 10
11 + ((3 / 1) + 10)
(11 * 3) + (1 - 10)
((11 * 3) + 1) - 10
(11 + 3) + (10 * 1)
((11 + 3) + 10) * 1
(11 + (3 + 10)) * 1
11 + ((3 + 10) * 1)
11 + (3 + (10 * 1))
(11 + 3) + (10 / 1)
((11 + 3) + 10) / 1
(11 + (3 + 10)) / 1
11 + ((3 + 10) / 1)
11 + (3 + (10 / 1))
((11 * 3) - 10) + 1
(11 * 3) - (10 - 1)
(10 + (1 * 11)) + 3
10 + ((1 * 11) + 3)
10 + (1 * (11 + 3))
(10 * 1) + (11 + 3)
((10 * 1) + 11) + 3
(10 * 1) + (3 + 11)
((10 * 1) + 3) + 11
((11 * 3) - 10) + 1
(11 * 3) - (10 - 1)
(10 + (1 * 11)) + 3
10 + ((1 * 11) + 3)
10 + (1 * (11 + 3))
(10 * 1) + (11 + 3)
((10 * 1) + 11) + 3
(10 / 1) + (11 + 3)
((10 / 1) + 11) + 3
(10 + (1 * 3)) + 11
10 + ((1 * 3) + 11)
10 + (1 * (3 + 11))
(10 * 1) + (3 + 11)
((10 * 1) + 3) + 11
(10 / 1) + (3 + 11)
((10 / 1) + 3) + 11
(10 + 11) + (1 * 3)
10 + (11 + (1 * 3))
((10 + 11) * 1) + 3
(10 + (11 * 1)) + 3
10 + ((11 * 1) + 3)
((10 + 11) / 1) + 3
(10 + (11 / 1)) + 3
10 + ((11 / 1) + 3)
(10 + 11) + (3 * 1)
((10 + 11) + 3) * 1
(10 + (11 + 3)) * 1
10 + ((11 + 3) * 1)
10 + (11 + (3 * 1))
(10 + 11) + (3 / 1)
((10 + 11) + 3) / 1
(10 + (11 + 3)) / 1
10 + ((11 + 3) / 1)
10 + (11 + (3 / 1))
(10 + 3) + (1 * 11)
10 + (3 + (1 * 11))
((10 + 3) * 1) + 11
(10 + (3 * 1)) + 11
10 + ((3 * 1) + 11)
((10 + 3) / 1) + 11
(10 + (3 / 1)) + 11
10 + ((3 / 1) + 11)
(10 + 3) + (11 * 1)
((10 + 3) + 11) * 1
(10 + (3 + 11)) * 1
10 + ((3 + 11) * 1)
10 + (3 + (11 * 1))
(10 + 3) + (11 / 1)
((10 + 3) + 11) / 1
(10 + (3 + 11)) / 1
10 + ((3 + 11) / 1)
10 + (3 + (11 / 1))
(3 + (1 * 11)) + 10
3 + ((1 * 11) + 10)
3 + (1 * (11 + 10))
(3 * 1) + (11 + 10)
((3 * 1) + 11) + 10
3 + ((11 * 1) + 10)
((3 + 11) / 1) + 10
(3 + (11 / 1)) + 10
```

```
((10 + 3) + 11) * 1
(10 + (3 + 11)) * 1
10 + ((3 + 11) * 1)
10 + (3 + (11 * 1))
(10 + 3) + (11 / 1)
((10 + 3) + 11) / 1
(10 + (3 + 11)) / 1
10 + ((3 + 11) / 1)
10 + (3 + (11 / 1))
(3 + (1 * 11)) + 10
3 + ((1 * 11) + 10)
3 + (1 * (11 + 10))
(3 * 1) + (11 + 10)
((3 * 1) + 11) + 10
3 + ((11 * 1) + 10)
((3 + 11) / 1) + 10
(3 + (11 / 1)) + 10
3 + ((11 / 1) + 10)
(3 * 11) + (1 - 10)
((3 * 11) + 1) - 10
(3 + 11) + (10 * 1)
((3 + 11) + 10) * 1
(3 + (11 + 10)) * 1
3 + ((11 + 10) * 1)
3 + (11 + (10 * 1))
(3 + 11) + (10 / 1)
((3 + 11) + 10) / 1
(3 + (11 + 10)) / 1
3 + ((11 + 10) / 1)
3 + (11 + (10 / 1))
((3 * 11) - 10) + 1
(3 * 11) - (10 - 1)
(3 + 10) + (1 * 11)
3 + (10 + (1 * 11))
((3 + 10) * 1) + 11
(3 + (10 * 1)) + 11
3 + ((10 * 1) + 11)
((3 + 10) / 1) + 11
(3 + (10 / 1)) + 11
3 + ((10 / 1) + 11)
(3 + 10) + (11 * 1)
((3 + 10) + 11) * 1
(3 + (10 + 11)) * 1
3 + ((10 + 11) * 1)
3 + (10 + (11 * 1))
(3 + 10) + (11 / 1)
((3 + 10) + 11) / 1
(3 + (10 + 11)) / 1
3 + ((10 + 11) / 1)
3 + (10 + (11 / 1))
Apakah ingin menyimpan solusi? (Y/N)
Y
Masukkan Nama File:
no5rand
File created and write succesfully
Execution time: 65 ms
PS C:\Users\MSI\OneDrive\Dokumen\Kuliah\Kuliah Semester 4\Strategi Algoritma\Tucil1_13521076>
```

6) A A A 2

```
PS C:\Users\MSI\OneDrive\Dokumen\Kuliah\Kuliah Semester 4\Strategi Algoritma\Tucil1_13521076>  c:; cd 'c:\Users\MSI\OneDrive
\Dokumen\Kuliah\Kuliah Semester 4\Strategi Algoritma\Tucil1_13521076'; & 'C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe' '
-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\MSI\AppData\Roaming\Code\User\workspaceStorage\9daae880b65f6bd9de10
a9a066e81ec5\redhat.java\jdt_ws\Tucil1_13521076_5bf31e26\bin' 'main'
Pilih masukkan yang diinginkan:
1. Random
2. Keyboard Sendiri
2
Masukkan 4 angka/huruf (A,2,3,4,5,6,7,8,9,10,J,Q,K)
A
A
A
2
0 Solutions found
Tidak ada solusi
Apakah ingin menyimpan solusi? (Y/N)
Y
Masukkan Nama File:
no6nosolution
File created and write succesfully
Execution time: 7 ms
PS C:\Users\MSI\OneDrive\Dokumen\Kuliah\Kuliah Semester 4\Strategi Algoritma\Tucil1_13521076>
```

LINK REPOSITORY

https://github.com/AghnaAbyan/Tucil1_13521076

CHECKLIST

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan		
2. Program berhasil *running*		
3. Program dapat membaca input/generate sendiri dan memberikan luaran		
4. Solusi yang diberikan program memenuhi (berhasil mencapai 24)		
5. Program dapat menyimpan solusi dalam file teks		